Comparing Local and Sequential Models for Statistical Incremental Natural Language Understanding

Silvan Heintze, Timo Baumann, David Schlangen
Department of Linguistics
University of Potsdam, Germany
firstname.lastname@uni-potsdam.de

Abstract

Incremental natural language understanding is the task of assigning semantic representations to successively larger prefixes of utterances. We compare two types of statistical models for this task: a) *local models*, which predict a single class for an input; and b), *sequential models*, which align a sequence of classes to a sequence of input tokens. We show that, with some modifications, the first type of model can be improved and made to approximate the output of the second, even though the latter is more informative. We show on two different data sets that both types of model achieve comparable performance (significantly better than a baseline), with the first type requiring simpler training data. Results for the first type of model have been reported in the literature; we show that for our kind of data our more sophisticated variant of the model performs better.

1 Introduction

Imagine being at a dinner, when your friend Bert says "My friend, can you pass me the salt over there, please?". It is quite likely that you get the idea that something is wanted of you fairly early into the utterance, and understand what exactly it is that is wanted even before the utterance is over.

This is possible only because you form an understanding of the meaning of the utterance even before it is complete; an understanding which you refine—and possibly revise—as the utterance goes on. You understand the utterance *incrementally*. This is something that is out of reach for most current dialogue systems, which process utterances non-incrementally, *en bloc* (cf. (Skantze and Schlangen, 2009), *inter alia*).

Enabling incremental processing in dialogue systems poses many challenges (Allen et al., 2001; Schlangen and Skantze, 2009); we focus here on the sub-problem of modelling incremental understanding—a precondition for enabling truly interactive behaviour. More specifically, we look at statistical methods for learning mappings between (possibly partial) utterances and meaning representations. We distinguish between two types of understanding, which were sketched in the first paragraph above: a) forming a *partial* understanding, and b) *predicting* a complete understanding.

Recently, some results have been published on b), predicting utterance meanings, (Sagae et al., 2009; Schlangen et al., 2009). We investigate here how well this predictive approach works in two other domains, and how a simple extension of techniques (ensembles of slot-specific classifiers vs. one frame-specific one) can improve performance. To our knowledge, task a), computing partial meanings, has so far only been tackled with symbolic methods (e.g., (Milward and Cooper, 1994; Aist et al., 2006; Atterer and Schlangen, 2009));[1] we present here some first results on approaching it with statistical models.

Plan of the paper: First, we discuss relevant previous work. We then define the task of incremental natural language understanding and its two variants in more detail, also looking at how models can be evaluated. Finally, we present and discuss the results of our experiments, and close with a conclusion and some discussion of future work.

2 Related Work

Statistical natural language understanding is an active research area, and many sophisticated models for this task have recently been published, be that *generative* models (e.g., in (He and Young, 2005)), which learn a joint distribution over in-

[1] We explicitly refer to computation of incremental interpretations here; there is of course a large body of work on statistical incremental *parsing* (e.g., (Stolcke, 1995; Roark, 2001)).

(Mairesse et al., 2009)	94.50
(He and Young, 2005)	90.3
(Zettlemoyer and Collins, 2007)	95.9
(Meza et al., 2008)	91.56

Table 1: Recent published f-scores for non-incremental statistical NLU, on the ATIS corpus

put, output and possibly hidden variables; or, more recently, *discriminative* models (e.g., (Mairesse et al., 2009)) that directly learn a mapping between input and output. Much of this work uses the ATIS corpus (Dahl et al., 1994) as data and hence is directly comparable. In Table 1, we list the results achieved by this work; we will later situate our results relative to this.

That work, however, only looks at mappings between complete utterances and semantic representations, whereas we are interested in the process of mapping semantic representations to successively larger utterance fragments. More closely related then is (Sagae et al., 2009; DeVault et al., 2009), where a maximum entropy model is trained for mapping utterance fragments to semantic frames. (Sagae et al., 2009) make the observation that often the quality of the prediction does not increase anymore towards the end of the utterance; that is, the meaning of the utterance can be predicted before it is complete.

In (Schlangen et al., 2009), we presented a model that predicts incrementally a specific aspect of the meaning of a certain type of utterance, namely the intended referent of a referring expression; the similarity here is that the output is of the same type regardless of whether the input utterance is complete or not.

(DeVault et al., 2009) discuss how such 'mind reading' can be used interactionally in a dialogue system, e.g. for completing the user's utterance as an indication of the system's grounding state. While these are interesting uses, the approach is somewhat limited by the fact that it is incremental only on the input side, while the output does not reflect how 'complete' (or not) the input is. We will compare this kind of incremental processing in the next section with one where the output is incremental as well, and we will then present results from our own experiments with both kinds of incrementality in statistical NLU.

3 Task, Evaluation, and Data Sets

3.1 The Task

We have said that the task of incremental natural language understanding consists in the assignment of semantic representations to progressively more complete prefixes of utterances. This description can be specified along several aspects, and this yields different versions of the task, appropriate for different uses. One question is what the assigned representations are, the other is what exactly they are assigned to. We investigate these questions here abstractly, before we discuss the instantiations in the next sections.

Let's start by looking at the types of representations that are typically assigned to *full* utterances. A type often used in dialogue systems is the *frame*, an attribute value matrix. (The attributes are here typically called *slots*.) These frames are normally typed, that is, there are restrictions on which slots can (and must) occur together in one frame. The frames are normally assigned to the utterance as a whole and not to individual words.

In an incremental setting, where the input potentially consists of an incomplete utterance, choosing this type of representation and style of assignment turns the task into one of *prediction* of the utterance meaning. What we want our model to deliver is a guess of what the meaning of the utterance is going to be, even if we have only seen a prefix of the utterance so far; we will call this "whole-frame output" below.[2]

Another popular representation of semantics in applied systems uses semantic *tags*, i.e., markers of semantic role that are attached to individual parts of the utterance. Such a style of assignment is inherently 'more incremental', as it provides a way to assign meanings that represent only what has indeed been said so far, and does not make assumptions about what will be said. The semantic representation of the prefix simply contains all and only the tags assigned to the words in the prefix; this will be called "aligned output" below. To our knowledge, the potential of this type of representation (and the models that create them) for incremental processing has not yet been explored; we present our first results below.

Finally, there is a hybrid form of representation and assignment. If we allow the output frames to 'grow' as more input comes in (hence possibly violating the typing of the frames as they are expected for full utterances), we get a form of representation with a notion of 'partial semantics' (as

[2] In (Schlangen and Skantze, 2009), this type of incremental processing is called "input incremental", as only the input is incrementally enriched, while the output is always of the same type (but may increase in quality).

only that is represented for which there is evidence in what has already been seen), but without *direct* association of parts of the representation and parts of the utterance or utterance prefix.

3.2 Evaluation

Whole-Frame Output A straightforward metric is *Correctness*, which can take the values 1 (output is exactly as expected) or 0 (output is *not* exactly as expected). Processing a test corpus in this way, we get one number for each utterance prefix, and, averaging this number, one measurement for the whole corpus.

This can give us a first indication of the general quality of the model, but because it weighs the results for prefixes of all lengths equally, it cannot tell us much about how well the incremental processing worked. In actual applications, we presumably do not expect the model to be correct from the very first word on, but do expect it to get better the longer the available utterance prefix becomes. To capture this, we define two more metrics: *first occurrence* (FO), as the position (relative to the eventual length of the full utterance) where the response was correct first; and *final decision* (FD) as the position from which on the response stayed correct (which consequently can only be measured if indeed the response stays correct).[3] The difference between FO and FD then tells us something about the stability of hypotheses of the model.

In some applications, we may indeed only be able to do further processing with fully correct—or at least correctly typed—frames; in which case *correctness* and FO/FD on frames are appropriate metrics. However, sometimes even frames that are only partially correct can be of use, for example if specific system reactions can be tied to individual slots. To give us more insight about the quality of a model in such cases, we need a metric that is finer-grained than binary correctness. Following (Sagae et al., 2009), we can conceptualise our task as one of retrieval of slot/value pairs, and use *precision* and *recall* (and, as their combination, *f-score*) as metrics. As we will see, it will be informative to plot the development of this score over the course of processing the utterance.

For these kinds of evaluations, we need as a gold standard only one annotation per utterance, namely the final frame.

Aligned Output As sequence alignments have more structure—there is a linear order between the tags, and there is exactly one tag per input token—correctness is a more fine-grained, and hence more informative, metric here; we define it as the proportion of tags that are correct in a sequence. We can also use precision and recall here, looking at each position in the sequence individually: Has the tag been recalled (true positive), or has something else been predicted instead (false negative, *and* false positive)? Lastly, we can also reconstruct frames from the tag sequences, where sequences of the same tag are interpreted as segmenting off the slot value. (And hence, what was several points for being right or wrong, one for each tag, becomes one, being either the correct slot value or not. We will discuss these differences when we show evaluations of aligned output.)

For this type of evaluation, we need gold-standard information of the same kind, that is, we need aligned tag sequences. This information is potentially more costly to create than the one final semantic representation needed for the whole-frame setting.

Hybrid Output As we will see below, the hybrid form of output ('growing' frames) is produced by ensembles of local classifiers, with one classifier for each possible slot. How this output can be evaluated depends on what type of information is available. If we only have the final frame, we can calculate f-score (in the hope that *precision* will be better than for the whole-frame classifier, as such a classifier ensemble can focus on predicting slots/value pairs for which there is direct evidence); if we do have sequence information, we can convert it to growing frames and evaluate against that.

3.3 The Data Sets

ATIS As our first dataset, we use the ATIS air travel information data (Dahl et al., 1994), as preprocessed by (Meza et al., 2008) and (He and Young, 2005). That is, we have available for each utterance a semantic frame as in (1), and also a tag sequence that aligns semantic concepts (same as the slot names) and words. One feature to note here about the ATIS representations is that the slot values / semantic atoms are just the words in the utterance. That is, the word itself is its own semantic representation, and no additional abstrac-

[3] These metrics of course can only be computed post-hoc, as during processing we do not know how long the utterance is going to be.

tion is performed. In this domain, this is likely unproblematic, as there aren't many different ways (that are to be expected in this domain) to refer to a given city or a day of the week, for example.

(1) "What flights are there arriving in Chicago after 11pm?"

$$\begin{bmatrix} \text{GOAL} = \text{FLIGHT} \\ \text{TOLOC.CITY\_NAME} = \text{Chicago} \\ \text{ARRIVE\_TIME.TIME\_RELATIVE} = \text{after} \\ \text{ARRIVE\_TIME.TIME} = \text{11pm} \end{bmatrix}$$

In our experiments, we use the ATIS training set which contains 4481 utterances, between 1 and 46 words in length (average 11.46; sd 4.34). The vocabulary consists of 897 distinct words. There are 3159 distinct frames, 2594 (or 58% of all frames) of which occur only once. Which of the 96 possible slots occur in a given frame is distributed very unevenly; there are some very frequent slots (like FROMLOC.CITYNAME or DEPART_DATE.DAY_NAME) and some very rare or even unique ones (e.g., ARRIVE_DATE.TODAY_RELATIVE, or TIME_ZONE).

Pentomino The second corpus we use is of utterances in a domain that we have used in much previous work (e.g., (Schlangen et al., 2009; Atterer and Schlangen, 2009; Fernández and Schlangen, 2007)), namely, instructions for manipulating puzzle pieces to form shapes. The particular version we use here was collected in a Wizard-of-Oz study, where the goal was to instruct the computer to pick up, delete, rotate or mirror puzzle tiles on a rectangular board, and drop them on another one. The user utterances were annotated with semantic frames and also aligned with tag sequences. We use here a frame representation where the slot value is a part of the utterance (as in ATIS), an example is shown in (2). (The corpus is in German; the example is translated here for presentation.) We show the full frame here, with all possible slots; unused slots are filled with "empty". Note that this representation is somewhat less directly usable in this domain than for ATIS; in a practical system, we'd need some further module (rule-based or statistical) that maps such partial strings to their denotations, as this mapping is less obvious here than in the travel domain.

(2) "Pick up the W-shaped piece in the upper right corner"

$$\begin{bmatrix} \text{action} = \text{"pick up"} \\ \text{tile} = \text{"the W-shaped piece} \\ \quad \text{in the upper right corner"} \\ \text{field} = \text{empty} \\ \text{rotpar} = \text{empty} \\ \text{mirpar} = \text{empty} \end{bmatrix}$$

The corpus contains 1563 utterances, average length 5.42 words (sd 2.35), with a vocabulary of 222 distinct words. There are 964 distinct frames, with 775 unique frames.

In both datasets we use transcribed utterances and not ASR output, and hence our results present an upper bound on real-world performance.

4 Local Models: Support Vector Machines

In this section we report the results of our experiments with local classifiers, i.e. models which, given an input, predict one out of a set of classes as an answer. Such models are very naturally suited to the *prediction task*, where the semantics of the full utterance is treated as its class, which is to be predicted on the basis of what possibly is only a prefix of that utterance. We will also look at a simple modification, however, which enables such models to do something that is closer to the task of computing partial meanings.

4.1 Experimental Setup

For our experiments with local models, we used the implementations of support vector machines provided by the WEKA toolkit (Witten and Frank, 2005); as baseline we use a simple majority class predictor.[4]

We used the standard WEKA tools to convert the utterance strings into word vectors. Training was always done with the full utterance, but testing was done on prefixes of utterances; i.e., a sentence with 5 words would be one instance in training, but in a testing fold it would contribute 5 instances, one with one word, one with two words, and so on.[5] Because of this special way of testing the classifiers, and also because of the modifica-

[4] We tried other classifiers (C4.5, logistic regression, naive Bayes) as well, and found comparable performance on a development set. However, because of the high time costs (some models needed > 40 hours for training and testing on modern multi-CPU servers) we do not systematically compare performance and instead focus on SVMs. In any case, our interest here is not in comparing classification algorithms, but rather in exploring approaches to the novel problem of statistical incremental NLU.

[5] On a development set, we tried training on utterance prefixes, but that degraded performance, presumably due to increase in ambiguous training instances (same beginnings of what ultimately are very different utterances).

tions described below, we had to provide our own methods for cross-validation and evaluation. For the larger ATIS data set, we used 10 folds in cross validation, and for the Pentomino dataset 20 folds.

4.2 Results

To situate our results, we begin by looking at the performance of the models that predict a **full frame**, when given a **full utterance**; this is the normal, "non-incremental" statistical NLU task.[6]

(3)

classf.	metric	ATIS	Pento
maj	correctness	1.07	1.79
maj	f-score	35.98	16.15
SVM	correctness	16.21	38.77
SVM	f-score	68.17	63.23

We see that the results for ATIS are considerably lower than the state of the art in statistical NLU (Table 1). This need not concern us too much here, as we are mostly interested in the dynamics of the incremental process, but it indicates that there is room for improvement with more sophisticated models and feature design. (We will discuss an example of an improved model shortly.) We also see a difference between the corpora reflected in these results: being *exactly* right (good correctness) seems to be harder on the ATIS corpus, while being *somewhat* right (good f-score) seems to be harder on the pento corpus; this is probably due to the different sizes of the search space of possible frame types (large for ATIS, small for pento).

What we are really interested in, however, is the performance when given only a **prefix of an utterance**, and how this develops over the course of processing successively larger prefixes. We can investigate this with Figure 1. First, look at the solid lines. The black line shows the average f-score at various prefix lengths (in 10% steps) for the ATIS data, the grey line for the pento corpus. We see that both lines show a relatively steady incline, meaning that the f-score continues to improve when more of the utterance is seen. This is interesting to note, as both (DeVault et al., 2009) and (Atterer et al., 2009) found that in their data, all that is to be known can often be found somewhat before the end of the utterance. That this does not work so well here is most likely due to the difference in domain and the resulting utterances. Utterances giving details about travel plans

[6]The results for ATIS are based on half of the overall ATIS data, as cross-validating the model on all data took prohibitively long, presumably due to the large number of unique frames / classes.

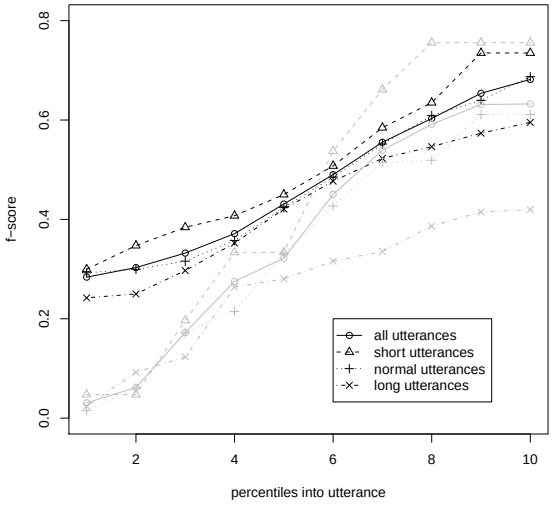

Figure 1: F-Score by Length of Prefix

are likely to present many important details, and some of them late into the utterance; cf. (1) above. The data from (DeVault et al., 2009) seems to be more conversational in nature, and, more importantly, presumable the expressible goals are less closely related to each other and hence can be read off of shorter prefixes.

As presented so far, the results are not very helpful for practical applications of incremental NLU. One thing one would like to know in a practical situation is how much the prediction of the model can be trusted for a given partial utterance. We would like to read this off graphs like those in the Figure—but of course, normally we cannot know what percentage of an utterance we have already seen! Can we trust this averaged curve if we do not know what length the incoming utterance will have?

To investigate this question, we have binned the test utterances into three classes, according to their length: "normal", for utterances that are of average length $\pm$ half a standard deviation, and "short" for all that are shorter, and "long" for all that are longer. The f-score curves for these classes are shown with the non-solid lines in Figure 1. We see that for ATIS there is not much variation compared to averaging over all utterances, and moreover, that the "normal" class very closely follows the general curve. On the pento data, the model seems to be comparably better for short utterances.

In a practical application, one could go with the assumption that the incoming utterance is going to be of normal length, and use the "normal"

curve for guidance; or one could devise an additional classifier that predicts the length-class of the incoming utterance, or more generally predicts whether a frame can already be trusted (DeVault et al., 2009). We leave this for future work.

As we have seen, the models that treat the semantic frame simply as a class label do not fare particularly well. This is perhaps not that surprising; as discussed above, in our corpora there aren't that many utterances with exact the same frame. Perhaps it would help to break up the task, and train **individual classifiers for each slot**?[7] This idea can be illustrated with (2) above. There we already included "unused" slots in the frame; if we now train classifiers for each slot, allowing them to predict "empty" in cases where a slot is unused, we can in theory reconstruct any frame from the ensemble of classifiers. To cover the pento data, the ensemble is small (there are 5 frames); it is considerably larger for ATIS, where there are so many distinct slots.

Again we begin by looking at the performance for **full utterances** (i.e., at 100% utterance length), but this time for **constructing the frame** from the reply of the classifier ensemble:

(4)

classf.	metric	ATIS	Pento
maj	correctness	0.16	0
maj	f-score	33.18	20.24
SVM	correctness	52.69	50.48
SVM	f-score	86.79	73.15

We see that this approach leads to an impressive improvement on the ATIS data (83.64 f-score instead of 68.17), whereas the improvement on the pento data is more modest (73.15 / 63.23).

Figure 2 shows the incremental development of the f-scores for the reconstructed frame. We see a similar shape in the curves; again a relatively steady incline for ATIS and a more dramatic shape for pento, and again some differences in behaviour for the different length classes of utterances. However, by just looking at the reconstructed frame, we are ignoring valuable information that the slot-classifier approach gives us. In some applications, we may already be able to do something useful with partial information; e.g., in the ATIS domain, we could look up an airport as soon as a FROMLOC becomes known. Hence, we'd want more fine-grained information, not just about when we can trust the whole frame, but rather about when

[7] A comparable approach is used for the non-incremental case for example by (Mairesse et al., 2009).

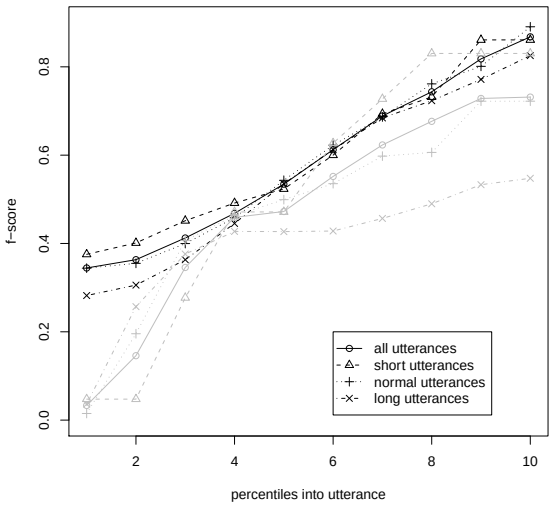

Figure 2: F-Score by Length of Prefix; Slot Classifiers

we can trust individual predicted slot values. (And so we move from the *prediction* task to the *partial representations* task.)

To explore this, we look at *First Occurrence* and *Final Decision* for some selected slots in Table 2. For some slots, the first occurrence (FO) of the correct value comes fairly early into the utterance (e.g., for the name of the airline it's at ca. 60%, for the departure city at ca. 63%, both with relatively high standard deviation, though) while others are found the first time rather late (goal city at 81%). This conforms well with intuitions about how such information would be presented in an utterance ("I'd like to fly on Lufthansa from Berlin to Tokyo").

We also see that the predictions are fairly stable: the number of cases where the slot value stays correct until the end is almost the same as that where it is correct at least once (FD *appl*icable vs. FO apl), and the average position is almost the same. In other words, the classifiers seem to go fairly reliably from "empty" (no value) to the correct value, and then seem to stay there. The overhead of unnecessary edits (EO) is fairly low for all slots shown in the table. (Ideally, EO is 0, meaning that there is no change except the one from "empty" to correct value.) All this is good news, as it means that a later module in a dialogue system can often begin to work with the partial results as soon as a slot-classifier makes a non-empty prediction. In an actual application, how trustworthy the individual classifiers are would then be read off statistics

slot name	avg FO	stdDev	apl	avg FD	stdDev	apl	avg EO	stdDev	apl
AIRLINE_NAME	0.5914	0.2690	506	0.5909	0.2698	501	0.5180	0.5843	527
DEPART_TIME.PERIOD_OF_DAY	0.7878	0.2506	530	0.7992	0.2476	507	0.2055	0.5558	579
FLIGHT_DAYS	0.4279	0.2660	37	0.4279	0.2660	37	0.0000	0.0000	37
FROMLOC.CITY_NAME	0.6345	0.1692	3633	0.6368	0.1692	3554	0.1044	0.4526	3718
ROUND_TRIP	0.5366	0.2140	287	0.5366	0.2140	287	0.0104	0.1015	289
TOLOC.CITY_NAME	0.8149	0.1860	3462	0.8162	0.1856	3441	0.2348	0.5723	3628
frames	0.9745	0.0811	2382	0.9765	0.0773	2361	0.7963	1.1936	4481

Table 2: FO/FD/EO for some selected slots; averaged over utterances of all lengths

like these, given a corpus from the domain.

To conclude this section, we have shown that classifiers that predict a complete frame based on utterance prefixes have a somewhat hard task here (harder, it seems, than in the corpus used in (Sagae et al., 2009), where they achieve an f-score of 87 on transcribed utterances), and the prediction results improve steadily throughout the whole utterance, rather than reaching their best value before its end. When the task is 'spread' over several classifiers, with each one responsible for only one slot, performance improves drastically, and also, the results become much more 'incremental'. We now turn to models that by design are more incremental in this sense.

5 Sequential Models: Conditional Random Fields

5.1 Experimental Setup

We use Conditional Random Fields (Lafferty et al., 2001) as our representative of the class of sequential models, as implemented in CRF++.[8] We use a simple template file that creates features based on a left context of three words.

Even though sequential models have the potential to be truly incremental (in the sense that they could produce a new output when fed a new increment, rather than needing to process the whole prefix again), CRF++ is targeted at tagging applications, and expects full sequences. We hence test in the same way as the SVMs from the previous section, by computing a new tag sequence for each prefix. Training again is done only on full utterances / tag sequences.

We compare the CRF results against two baselines. The simplest consists of just always choosing the most frequent tag, which is "O" (for *other*, marking material that does not contribute directly to the relevant meaning of the utterance, such as "please" in "I'd like to return on Monday, please."). The other baseline tags each word with its most frequent training data tag.

[8] http://crfpp.sourceforge.net/

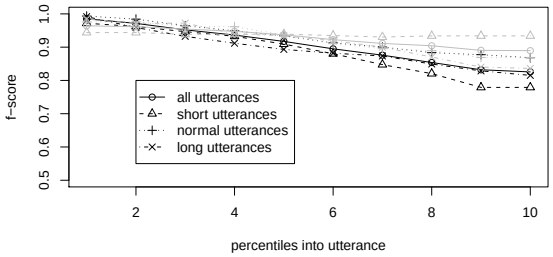

Figure 3: F-Score by Length of Prefix

ATIS	Corr.	Tag F-Score	Frame F-Score
CRF	93.38	82.56	76.10
Maj	85.14	60.86	48.08
O	63.43	00.31	00.31
Pento	Corr.	Tag F-Score	Frame F-Score
CRF	89.19	88.95	76.94
Maj	80.20	80.13	65.94
O	5.90	0.19	0.19

Table 3: Results of CRF models

5.2 Results

We again begin by looking at the limiting case, the results for **full utterances** (i.e., at the 100% mark).

Table 3 show three sets of results for each corpus. *Correctness* looks at the proportion of tags in a sequence that were correct. This measure is driven up by correct recognition of the dummy tag "o"; as we can see, this is quite frequently correct in ATIS, which drives up the "always use O"-baseline. Tag F-Score values the important tags higher; we see here, though, that the majority baseline (each word tagged with its most frequent tag) is surprisingly good. It is solidly beaten for the ATIS data, though. On the pento data, with its much smaller tagset (5 as opposed to 95), this baseline comes very high, but still the learner is able to get some improvement. The last metric evaluates reconstructed frames. It is stricter, because it offers less potential to be right (a sequence of the same tag will be translated into one slot value, turning several opportunities to be right into

only one).

The incremental dynamics looks quite different here. Since the task is not one of prediction, we do not expect to get better with more information; rather, we start at an optimal point (when nothing is said, nothing can be wrong), and hope that we do not amass too many errors along the way. Figure 3 confirms this, showing that the classifier is better able to keep the quality for the pento data than for the ATIS data. Also, there is not much variation depending on the length of the utterance.

6 Conclusions

We have shown how sequential and local statistical models can be used for two variants of the incremental NLU task: prediction, based on incomplete information, and assignment of partial representations to partial input. We have shown that breaking up the prediction task by using an ensemble of classifiers improves performance, and creates a hybrid task that sits between prediction and incremental interpretation.

While the objective quality as measured by our metrics is quite good, what remains to be shown is how such models can be integrated into a dialogue system, and how what they offer can be turned into improvements on interactivity. This is what we are turning to next.

Acknowledgements Funded by ENP grant from DFG.

References

G.S. Aist, J. Allen, E. Campana, L. Galescu, C.A. Gomez Gallo, S. Stoness, M. Swift, and M Tanenhaus. 2006. Software architectures for incremental understanding of human speech. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pittsburgh, PA, USA, September.

James Allen, George Ferguson, and Amanda Stent. 2001. An architecture for more realistic conversational systems. In *Proceedings of the conference on intelligent user interfaces*, Santa Fe, USA, June.

Michaela Atterer and David Schlangen. 2009. RUBISC – a robust unification-based incremental semantic chunker. In *Proceedings of the 2nd International Workshop on Semantic Representation of Spoken Language (SRSL 2009)*, Athens, Greece, March.

Michaela Atterer, Timo Baumann, and David Schlangen. 2009. No sooner said than done? testing incrementality of semantic interpretations of spontaneous speech. In *Proceedings of Interspeech 2009*, Brighton, UK, September.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: the atis-3 corpus. In *Proceedings of the workshop on Human Language Technology*, pages 43–48, Plainsboro, NJ, USA.

David DeVault, Kenji Sagae, and David Traum. 2009. Can i finish? learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL'09)*, London, UK, September.

Raquel Fernández and David Schlangen. 2007. Referring under restricted interactivity conditions. In Simon Keizer, Harry Bunt, and Tim Paek, editors, *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 136–139, Antwerp, Belgium, September.

Yulan He and Steve Young. 2005. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19(1):85–106.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289.

F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, and S. Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, April.

Ivan Meza, Sebastian Riedel, and Oliver Lemon. 2008. Accurate statistical spoken language understanding from limited development resources. In *In Proceedings of ICASSP*.

David Milward and Robin Cooper. 1994. Incremental interpretation: Applications, theory, and relationships to dynamic semantics. In *Proceedings of COLING 1994*, pages 748–754, Kyoto, Japan, August.

Brian Roark. 2001. *Robust Probabilistic Predictive Syntactic Processing: Motivations, Models, and Applications*. Ph.D. thesis, Department of Cognitive and Linguistic Sciences, Brown University.

Kenji Sagae, Gwen Christian, David DeVault, and David Traum. 2009. Towards natural language understanding of partial speech recognition results in dialogue systems. In *Short paper proceedings of the North American chapter of the Association for Computational Linguistics - Human Language Technologies conference (NAACL-HLT'09)*, Boulder, Colorado, USA, June.

David Schlangen and Gabriel Skantze. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 710–718, Athens, Greece, March.

David Schlangen, Timo Baumann, and Michaela Atterer. 2009. Incremental reference resolution: The task, metrics for evaluation, and a bayesian filtering model that is sensitive to disfluencies. In *Proceedings of SIGdial 2009, the 10th Annual SIGDIAL Meeting on Discourse and Dialogue*, London, UK, September.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 745–753, Athens, Greece, March.

Andreas Stolcke. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, USA, 2nd edition.

Luke S. Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of EMNLP-CoNLL*.